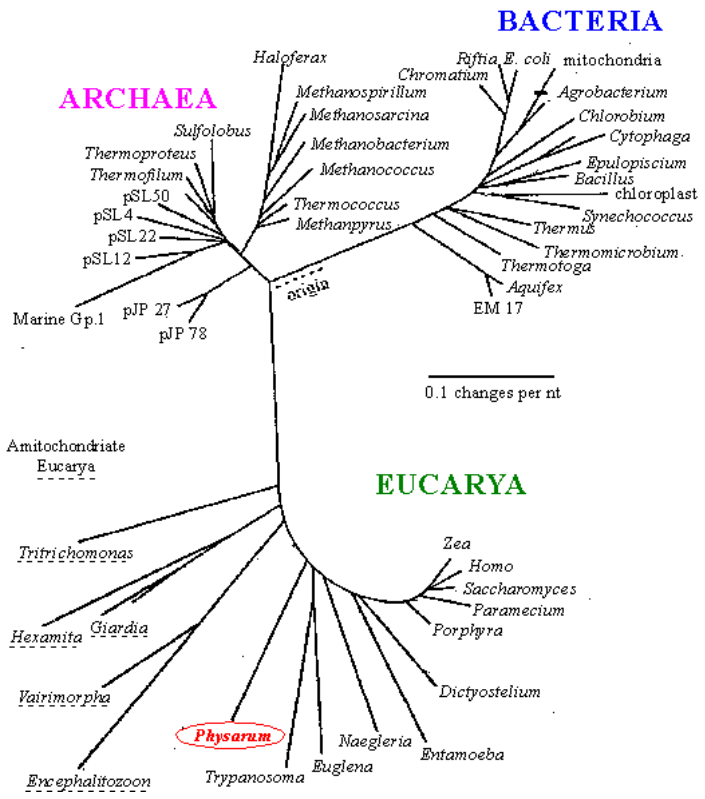
Pairwise Alignment and Database Searching

Anders Gorm Pedersen
Molecular Evolution Group
Center for Biological Sequence Analysis

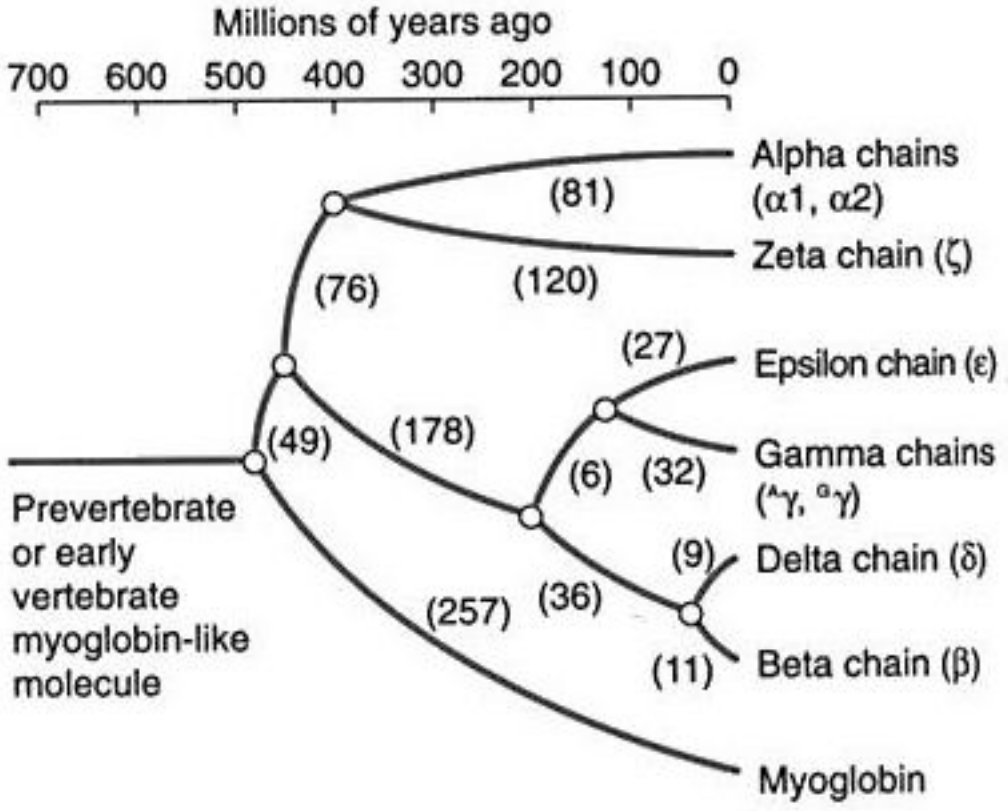
Sequences are related

- Darwin: all organisms are related through descent with modification
- => Sequences are related through descent with modification
- => Similar molecules have similar functions in different organisms



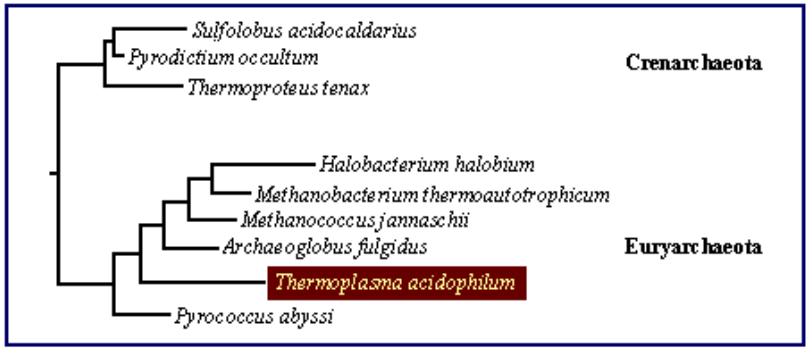
Phylogenetic tree based on ribosomal RNA:
three domains of life

Sequences are related, II



Phylogenetic tree of globin-type proteins found in humans

Why compare sequences?



- Determination of evolutionary relationships

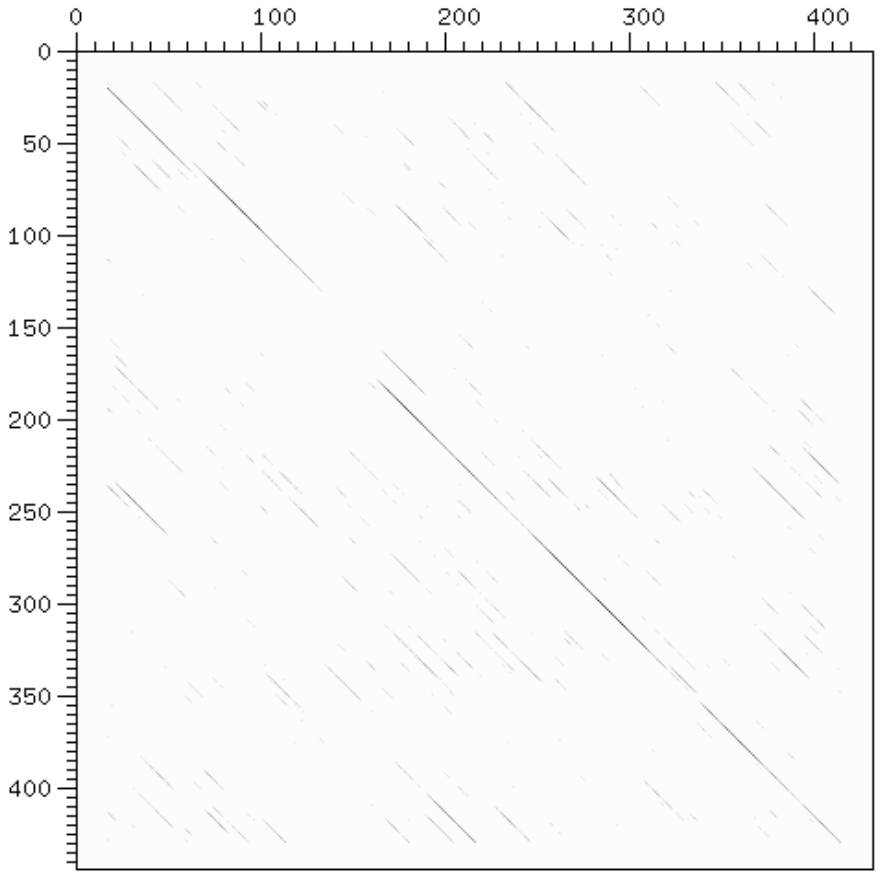
Protein 1: binds oxygen

Sequence similarity

Protein 2: binds oxygen ?

- Prediction of protein function and structure (database searches).

Dotplots: visual sequence comparison

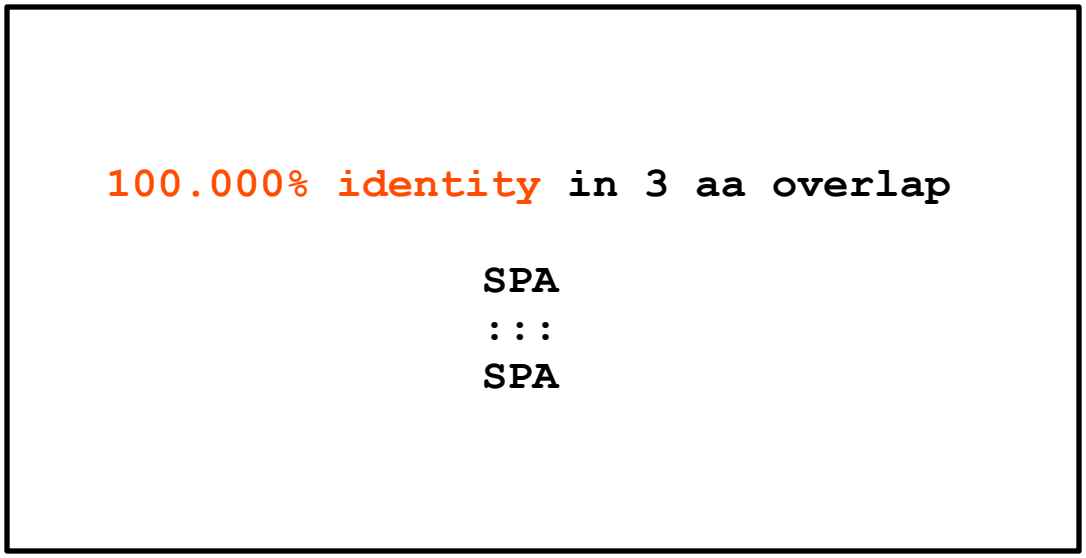


1. Place two sequences along axes of plot
2. Place dot at grid points where two sequences have identical residues
3. Diagonals correspond to conserved regions

Pairwise alignments

```
43.2% identity;                    Global alignment score: 374

                10         20         30         40                50
alpha  V-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS-----HGSA
       : :.: .:. : : ::::  .. : :.::: :... .: :. .:  : :::       :.
beta   VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNP
               10          20         30         40         50

            60         70         80         90        100       110
alpha  QVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHL
       .::.::::: :.....::.:.. .....::.::  ::.::: ::.::.. :. .:: :.
beta   KVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHF
       60         70         80         90        100       110

           120        130        140
alpha  PAEFTPAVHASLDKFLASVSTVLTSKYR
         :::: :.:. .: .:.:...:. ::.
beta   GKEFTPPVQAAYQKVVAGVANALAHKYH
      120        130        140
```

Pairwise alignment



Percent identity is not a good measure of alignment quality

Pairwise alignments: alignment score

```
43.2% identity;                    Global alignment score: 374

                10         20         30         40                50
alpha   V-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS-----HGSA
        : :.: .:. : : ::::  .. : :.::: :... .: :. .:  : :::       :.
beta    VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNP
               10          20         30         40         50

             60         70         80         90        100        110
alpha   QVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHL
        .::.::::: :.....::.:.. .....::.::  ::.::: ::.::.. :. .:: :.
beta    KVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHF
        60         70         80         90        100        110

            120        130        140
alpha   PAEFTPAVHASLDKFLASVSTVLTSKYR
          :::: :.:. .: .:.:...:. ::.
beta    GKEFTPPVQAAYQKVVAGVANALAHKYH
       120        130        140
```

Alignment scores: match vs. mismatch

Simple scoring scheme (too simple in fact…):

Matching amino acids: 5
Mismatch: 0

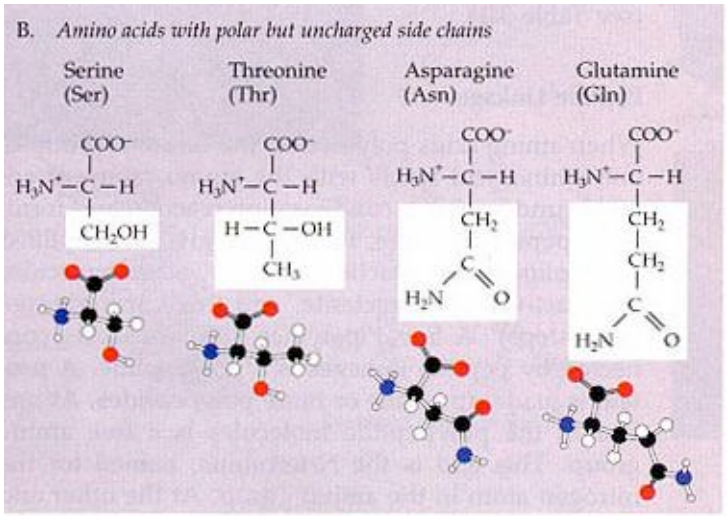
Scoring example:

```
K A W S A D V
:   : : :   :
K D W S A E V
5+0+5+5+5+0+5  =  25
```

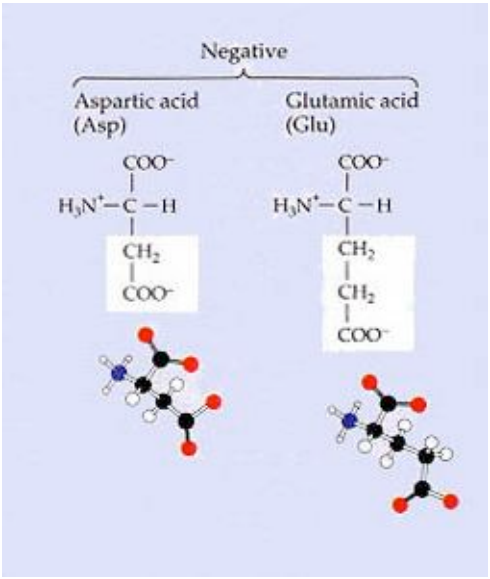
Pairwise alignments: conservative substitutions

```
43.2% identity;                    Global alignment score: 374


               10        20        30        40              50
alpha  V-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS-----HGSA
       : :.: .:. : : ::::  .. : :.::: :... .: :. .:  : :::       :.
beta   VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNP
              10          20        30        40        50

            60        70        80        90       100       110
alpha  QVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHL
       .::.::::: :.....::.:.. .....::.::  ::.::: ::.::.. :. .:: :.
beta   KVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHF
       60        70        80        90       100       110

           120       130       140
alpha  PAEFTPAVHASLDKFLASVSTVLTSKYR
         :::: :.:. .: .:.:...:. ::.
beta   GKEFTPPVQAAYQKVVAGVANALAHKYH
      120       130       140
```

Amino acid properties



Serine (S) and Threonine (T) have similar physicochemical properties



Aspartic acid (D) and Glutamic acid (E) have similar properties

=> Substitution of S/T or E/D occurs relatively often during evolution

=> Substitution of S/T or E/D should result in scores that are only moderately lower than identities

Protein substitution matrices

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	5																			
R	-2	7																		
N	-1	-1	7																	
D	-2	-2	**2**	**8**																
C	-1	-4	-2	-4	13															
Q	-1	1	0	0	-3	7														
E	-1	0	0	**2**	-3	**2**	**6**													
G	0	-3	0	-1	-3	-2	-3	8												
H	-2	0	**1**	-1	-3	**1**	0	-2	10											
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5										
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	**2**	5									
K	-1	**3**	0	-1	-3	**2**	**1**	-2	0	-3	-3	6								
M	-1	-2	-2	-4	-2	0	-2	-3	-1	**2**	**3**	-2	7							
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	**1**	-4	0	8						
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10					
S	**1**	-1	**1**	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	**5**				
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	**2**	**5**			
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	**1**	-4	-4	-3	15		
Y	-2	-1	-2	-3	-3	-1	-2	-3	**2**	-1	-1	-2	0	**4**	-3	-2	-2	**2**	8	
V	0	-3	-3	-4	-1	-3	-3	-4	-4	**4**	**1**	-3	**1**	-1	-3	-2	0	-3	-1	5

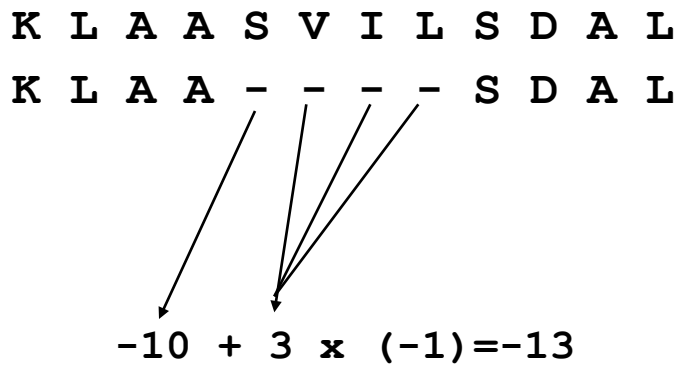
BLOSUM50 matrix:

- Positive scores on diagonal (identities)
- Similar residues get higher (positive) scores
- Dissimilar residues get smaller (negative) scores

Pairwise alignments: insertions/deletions

```
43.2% identity;                    Global alignment score: 374


               10        20        30        40              50
alpha  V-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS-----HGSA
       : :.: .:. : : ::::  .. : :.::: :... .: :. .:  : :::       :.
beta   VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNP
              10          20        30        40        50

            60        70        80        90       100       110
alpha  QVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHL
       .::.::::::  :.....::.:.. .....::.::  ::.::: ::.::.. :. .:: :.
beta   KVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHF
       60        70        80        90       100       110

           120       130       140
alpha  PAEFTPAVHASLDKFLASVSTVLTSKYR
         :::: :.:. .: .:.:...:. ::.
beta   GKEFTPPVQAAYQKVVAGVANALAHKYH
      120       130       140
```

Alignment scores: insertions/deletions



Affine gap penalties:

Multiple insertions/deletions may be one evolutionary event =>

Separate penalties for gap opening and gap elongation

Handout

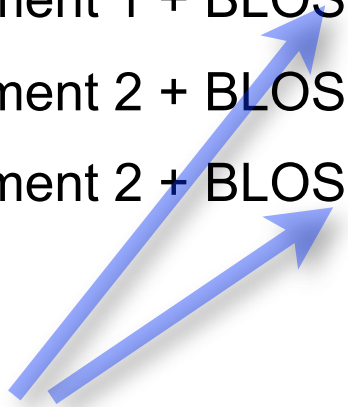
Compute 4 alignment scores: two different alignments using two different alignment matrices (and the same gap penalty system)

Score 1: Alignment 1 + BLOSUM-50 matrix + gaps

Score 2: Alignment 1 + BLOSUM-Trp matrix + gaps

Score 3: Alignment 2 + BLOSUM-50 matrix + gaps

Score 4: Alignment 2 + BLOSUM-Trp matrix + gaps



Note: fake matrix constructed for pedagogic purposes.

Handout: summary of results

	Alignment 1	Alignment 2
BLOSUM-50	38	51
BLOSUM-Trp	118	91

Protein substitution matrices

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	5																			
R	-2	7																		
N	-1	-1	7																	
D	-2	-2	2	8																
C	-1	-4	-2	-4	13															
Q	-1	1	0	0	-3	7														
E	-1	0	0	2	-3	2	6													
G	0	-3	0	-1	-3	-2	-3	8												
H	-2	0	1	-1	-3	1	0	-2	10											
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5										
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5									
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6								
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7							
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8						
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10					
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5				
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5			
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15		
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5

BLOSUM50 matrix:

- Positive scores on diagonal (identities)
- Similar residues get higher (positive) scores
- Dissimilar residues get smaller (negative) scores

Protein substitution matrices: different types

- **Identity matrix**
  (match vs. mismatch)

- **Genetic code matrix**
  (how similar are the codons?)

- **Chemical properties matrix**
  (use knowledge of physicochemical properties to design matrix)

- **Empirical matrices**
  (based on observed pair-frequencies in hand-made alignments)
  - PAM series
  - BLOSUM series
  - Gonnet

Estimation of the BLOSUM 50 matrix

- BLOSUM matrices are computed based on gap-free alignments in the so-called BLOCKS database. BLOSUM 50 is computed by comparing sequences that are less than 50% identical. BLOSUM 80 is computed from sequences less than 80% identical, etc.

- All pairs of sequences in a block are compared, and the observed pair frequencies are noted (e.g., A aligned with A makes up 1.5% of all pairs. A aligned with C makes up 0.01% of all pairs, etc.)

- Expected pair frequencies are computed from single amino acid frequencies. (e.g, $f_{A,C}=f_A$ x f_C=7% x 3% = 0.21%).

- For each amino acid pair the substitution scores are essentially computed as:

```
ID   FIBRONECTIN_2; BLOCK
COG9_CANFA  GNSAGEPCVFPFIFLGKQYSTCTREGRGDGHLWCATT
COG9_RABIT  GNADGAPCHFPFTFEGRSYTACTTDGRSDGMAWCSTT
FA12_HUMAN  LTVTGEPCHFPFQYHRQLYHKCTHKGRPGPQPWCATT
HGFA_HUMAN  LTEDGRPCRFPFRYGGRMLHACTSEGSAHRKWCATTH
MANR_HUMAN  GNANGATCAFPFKFENKWYADCTSAGRSDGWLWCGTT
MPRI_MOUSE  ETDDGEPCVFPFIYKGKSYDECVLEGRAKLWCSKTAN
PB1_PIG     AITSDDKCVFPFIYKGNLYFDCTLHDSTYYWCSVTTY
SFP1_BOVIN  ELPEDEECVFPFVYRNRKHFDCTVHGSLFPWCSLDAD
SFP3_BOVIN  AETKDNKCVFPFIYGNKKYFDCTLHGSLFLWCSLDAD
SFP4_BOVIN  AVFEGPACAFPFTYKGKKYYMCTRKNSVLLWCSLDTE
SP1_HORSE   AATDYAKCAFPFVYRGQTYDRCTTDGSLFRISWCSVT
COG2_CHICK  GNSEGAPCVFPFIFLGNKYDSCTSAGRNDGKLWCAST
COG2_HUMAN  GNSEGAPCVFPFTFLGNKYESCTSAGRSDGKMWCATT
COG2_MOUSE  GNSEGAPCVFPFTFLGNKYESCTSAGRNDGKVWCATT
COG2_RABIT  GNSEGAPCVFPFTFLGNKYESCTSAGRSDGKMWCATS
COG2_RAT    GNSEGAPCVFPFTFLGNKYESCTSAGRNDGKVWCATT
COG9_BOVIN  GNADGKPCVFPFTFQGRTYSACTSDGRSDGYRWCATT
COG9_HUMAN  GNADGKPCQFPFIFQGQSYSACTTDGRSDGYRWCATT
COG9_MOUSE  GNGEGKPCVFPFIFEGRSYSACTTKGRSDGYRWCATT
COG9_RAT    GNGDGKPCVFPFIFEGHSYSACTTKGRSDGYRWCATT
FINC_BOVIN  GNSNGALCHFPFLYNNHNYTDCTSEGRRDNMKWCGTT
FINC_HUMAN  GNSNGALCHFPFLYNNHNYTDCTSEGRRDNMKWCGTT
FINC_RAT    GNSNGALCHFPFLYSNRNYSDCTSEGRRDNMKWCGTT
MPRI_BOVIN  ETEDGEPCVFPFVFNGKSYEECVVESRARLWCATTAN
MPRI_HUMAN  ETDDGVPCVFPFIFNGKSYEECIIESRAKLWCSTTAD
PA2R_BOVIN  GNAHGTPCMFPFQYNQQWHHECTREGREDNLLWCATT
PA2R_RABIT  GNAHGTPCMFPFQYNHQWHHECTREGRQDDSLWCATT
```

$$\log \frac{\text{Pair-freq(obs)}}{\text{Pair-freq(expected)}} \qquad S_{A,C} = \log \frac{0.01\%}{0.21\%} = -1.3$$

Pairwise alignment

Optimal alignment:

alignment having the highest possible score given a substitution matrix and a set of gap penalties

Pairwise alignment: the problem

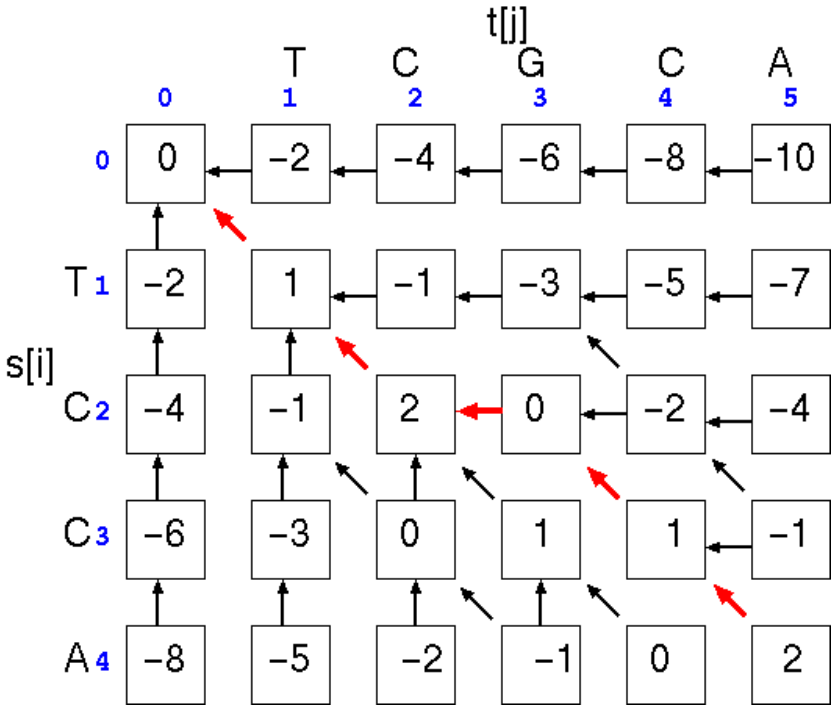
The number of possible pairwise alignments increases explosively with the length of the sequences:

Two protein sequences of length 100 amino acids can be aligned in approximately 10^{60} different ways

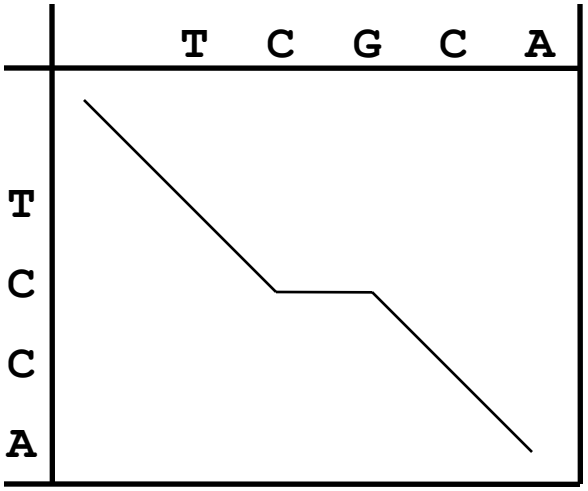
Time needed to test all possibilities is same order of magnitude as the entire lifetime of the universe.

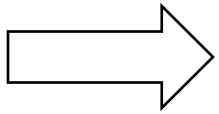
Pairwise alignment: the solution

"**Dynamic programming**"
(the Needleman-Wunsch algorithm)

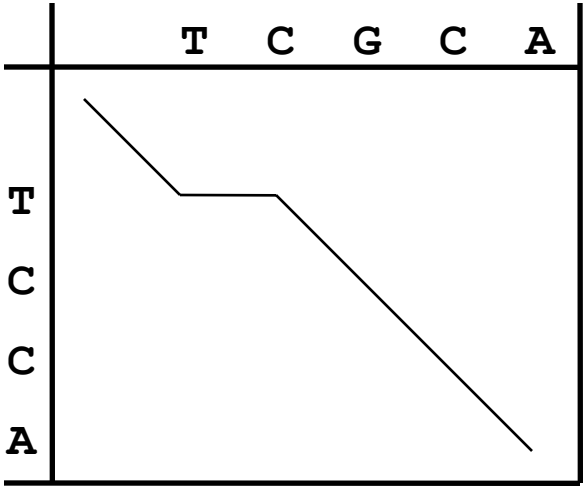


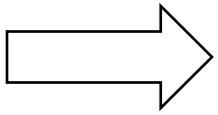
Alignment depicted as path in matrix





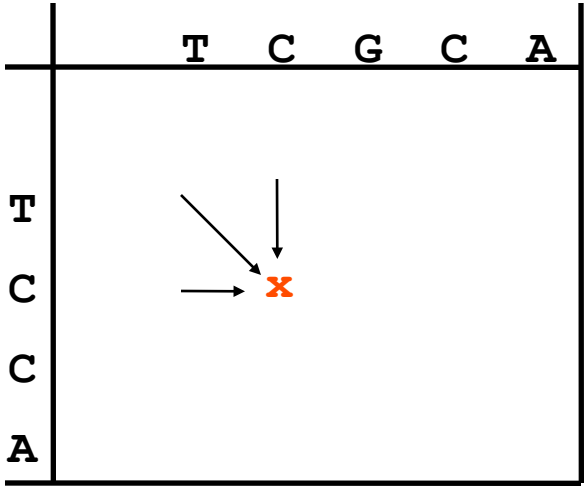
TCGCA
TC-CA





TCGCA
T-CCA

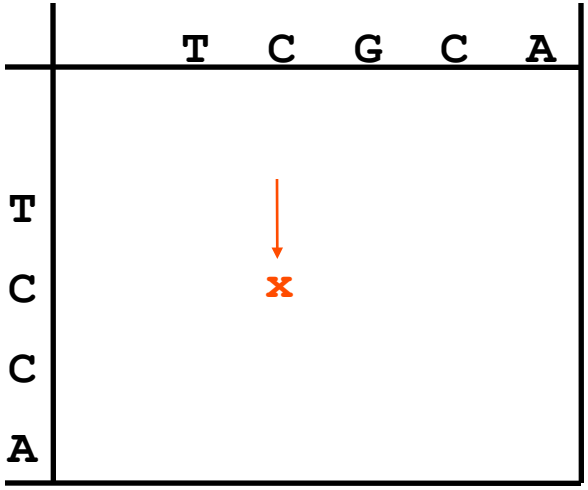
Dynamic programming: computation of scores



Any given point in matrix can only be reached from three possible previous positions (you cannot “align backwards”).

=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

Dynamic programming: computation of scores

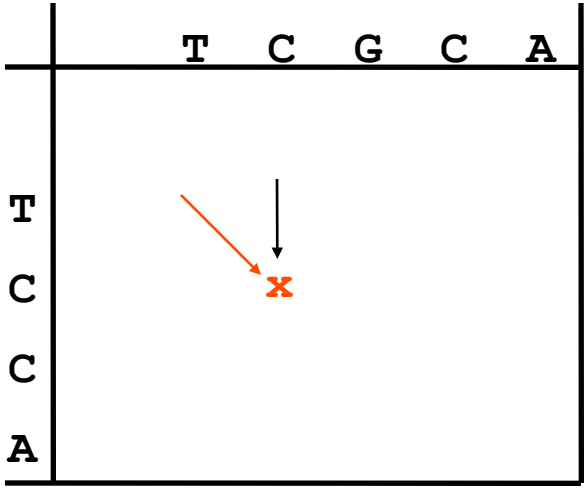


Any given point in matrix can only be reached from three possible positions (you cannot “align backwards”).

=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

score(x,y) = max { score(x,y-1) - gap-penalty

Dynamic programming: computation of scores

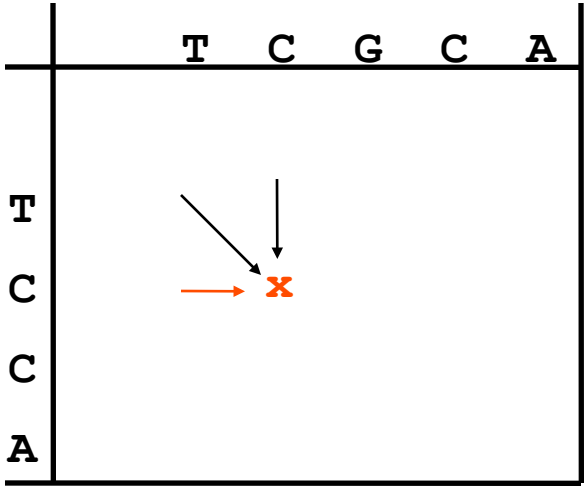


Any given point in matrix can only be reached from three possible positions (you cannot “align backwards”).

=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

$$\text{score}(x,y) = \max \begin{cases} \text{score}(x,y-1) - \text{gap-penalty} \\ \text{score}(x-1,y-1) + \text{substitution-score}(x,y) \end{cases}$$

Dynamic programming: computation of scores

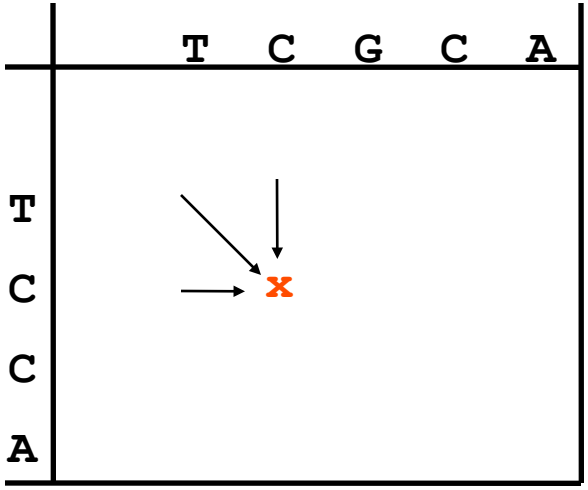


Any given point in matrix can only be reached from three possible positions (you cannot “align backwards”).

=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

$$\text{score}(x,y) = \max \begin{cases} \text{score}(x,y-1) - \text{gap-penalty} \\ \text{score}(x-1,y-1) + \text{substitution-score}(x,y) \\ \text{score}(x-1,y) - \text{gap-penalty} \end{cases}$$

Dynamic programming: computation of scores



Any given point in matrix can only be reached from three possible positions (you cannot “align backwards”).

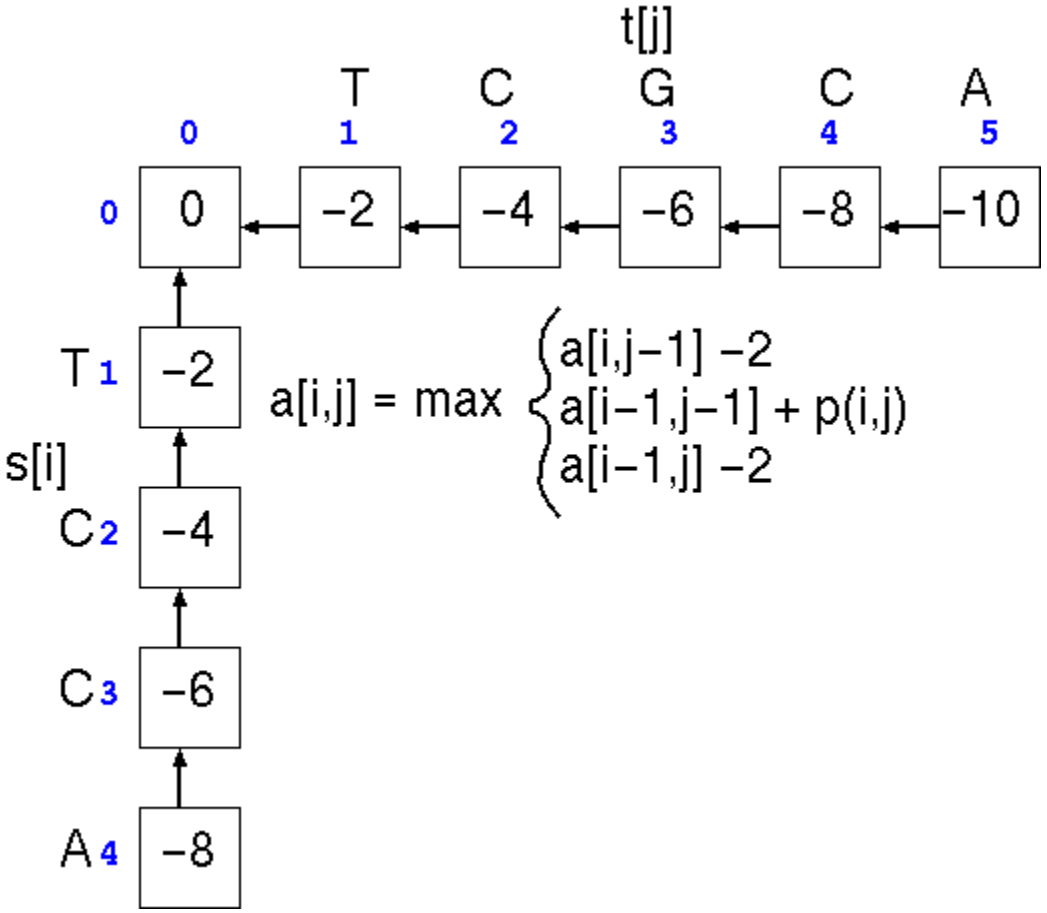
=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

Each new score is found by choosing the maximum of three possibilities. For each square in matrix: keep track of where best score came from.

Fill in scores one row at a time, starting in upper left corner of matrix, ending in lower right corner.

$$\text{score}(x,y) = \max \begin{cases} \text{score}(x,y-1) - \text{gap-penalty} \\ \text{score}(x-1,y-1) + \text{substitution-score}(x,y) \\ \text{score}(x-1,y) - \text{gap-penalty} \end{cases}$$

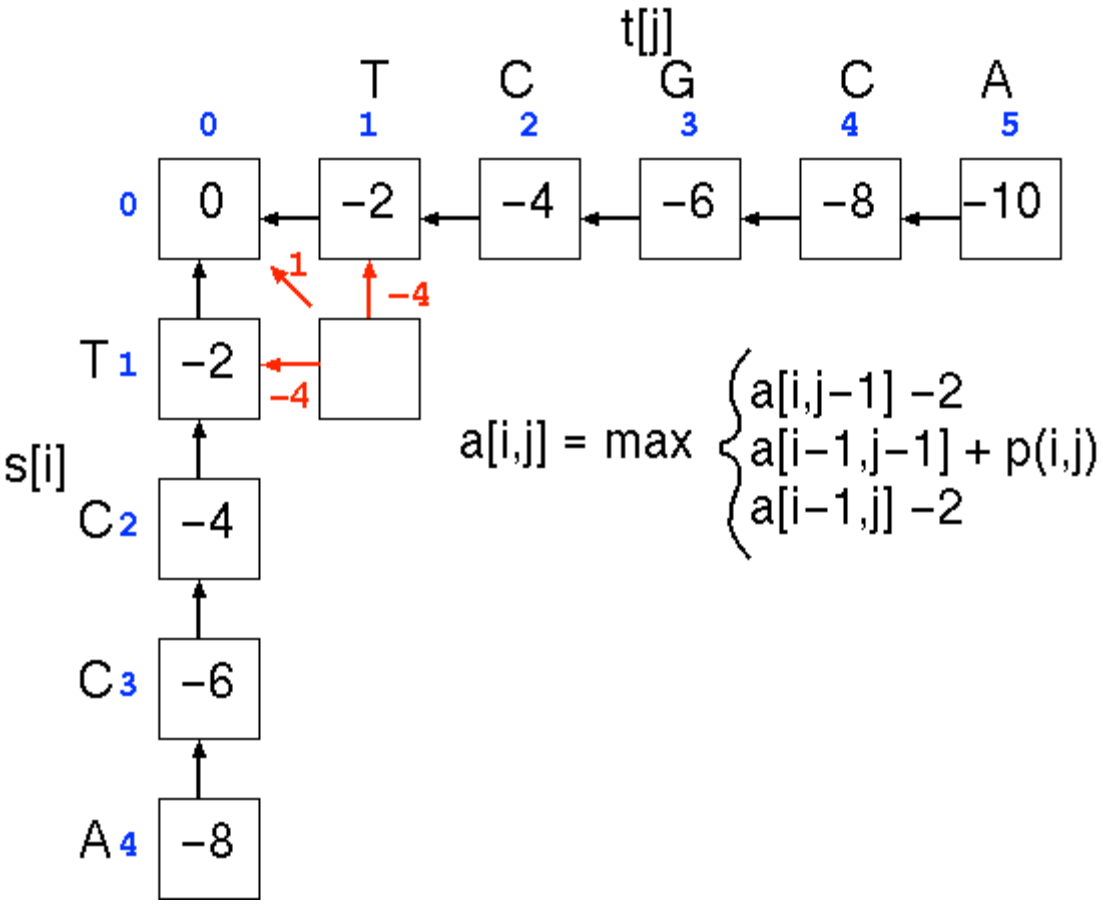
Dynamic programming: example



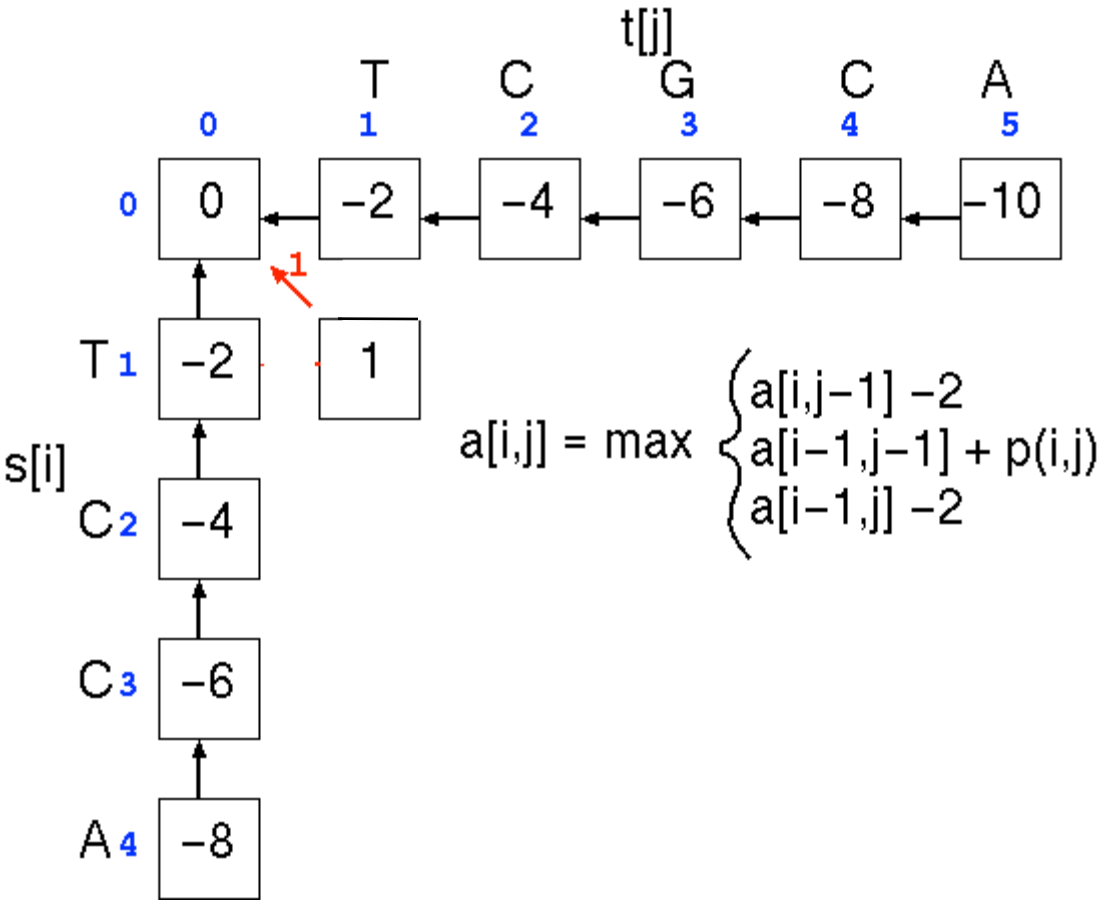
	A	C	G	T
A	1	-1	-1	-1
C	-1	1	-1	-1
G	-1	-1	1	-1
T	-1	-1	-1	1

Gaps: -2

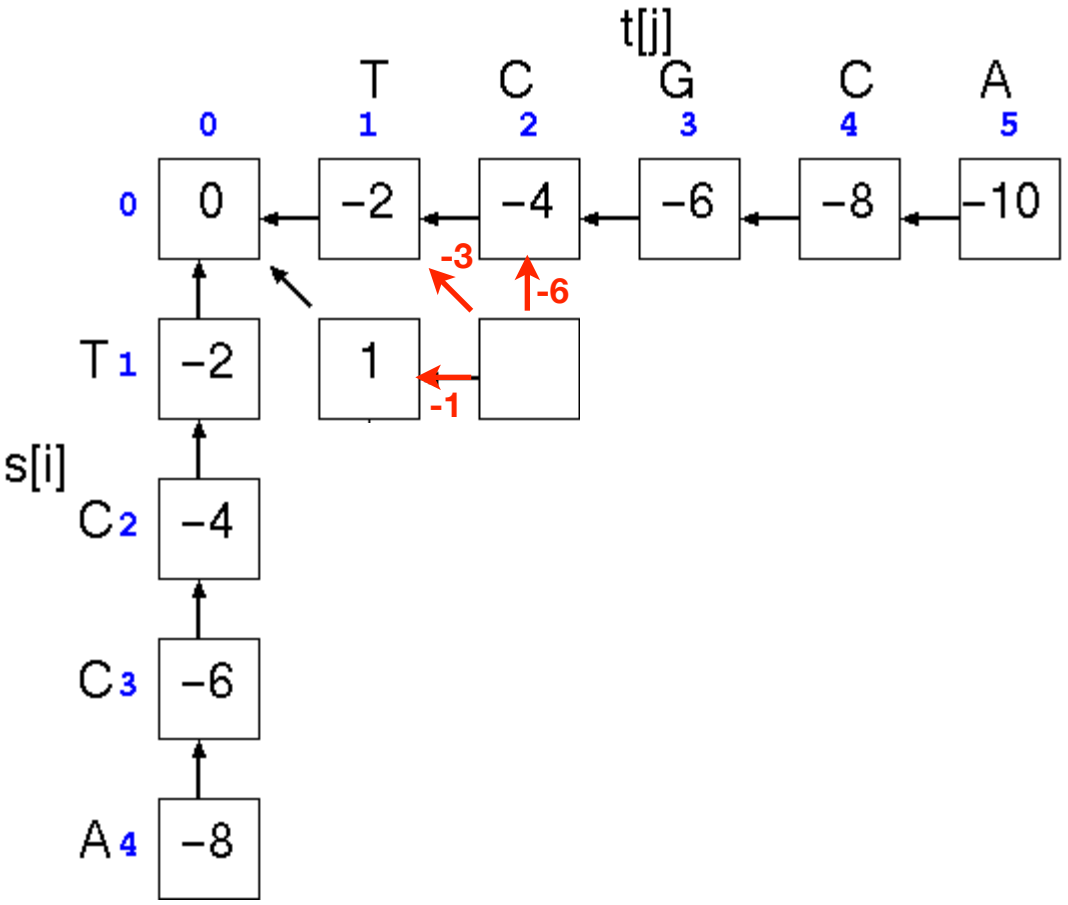
Dynamic programming: example



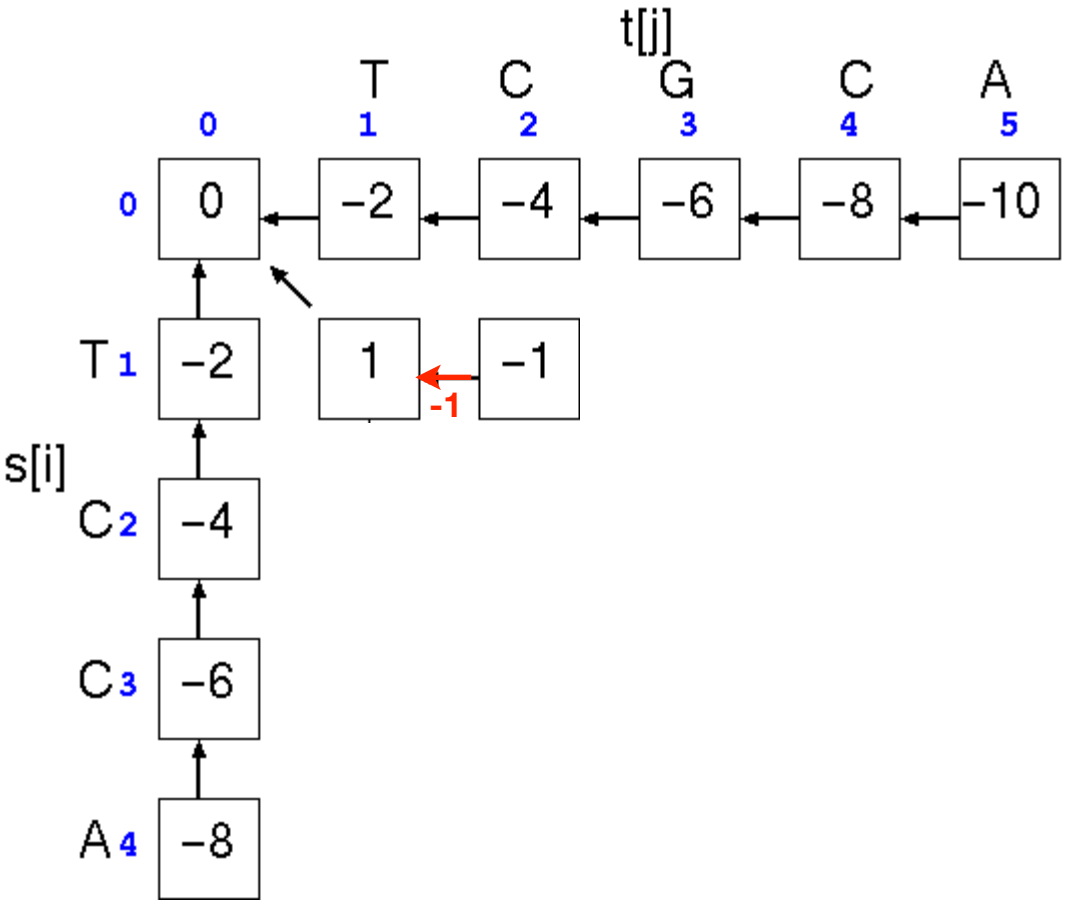
Dynamic programming: example



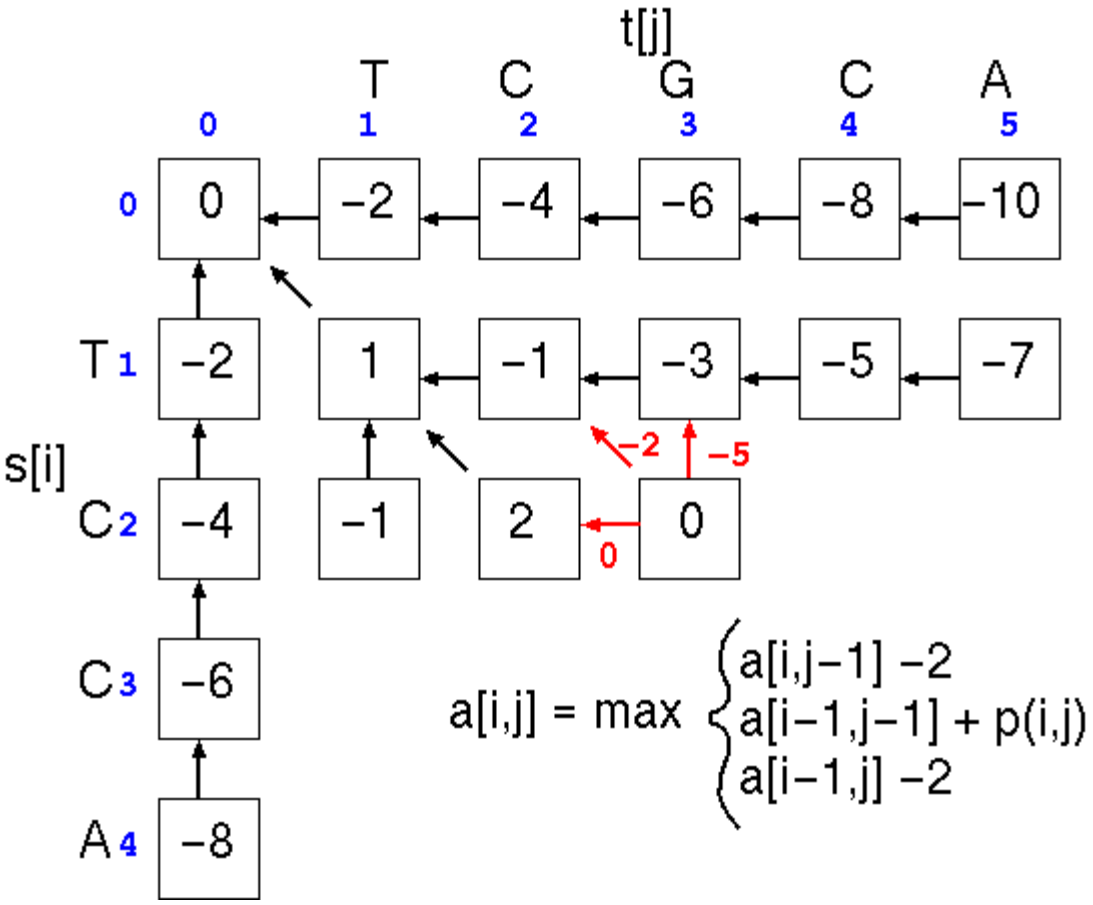
Dynamic programming: example



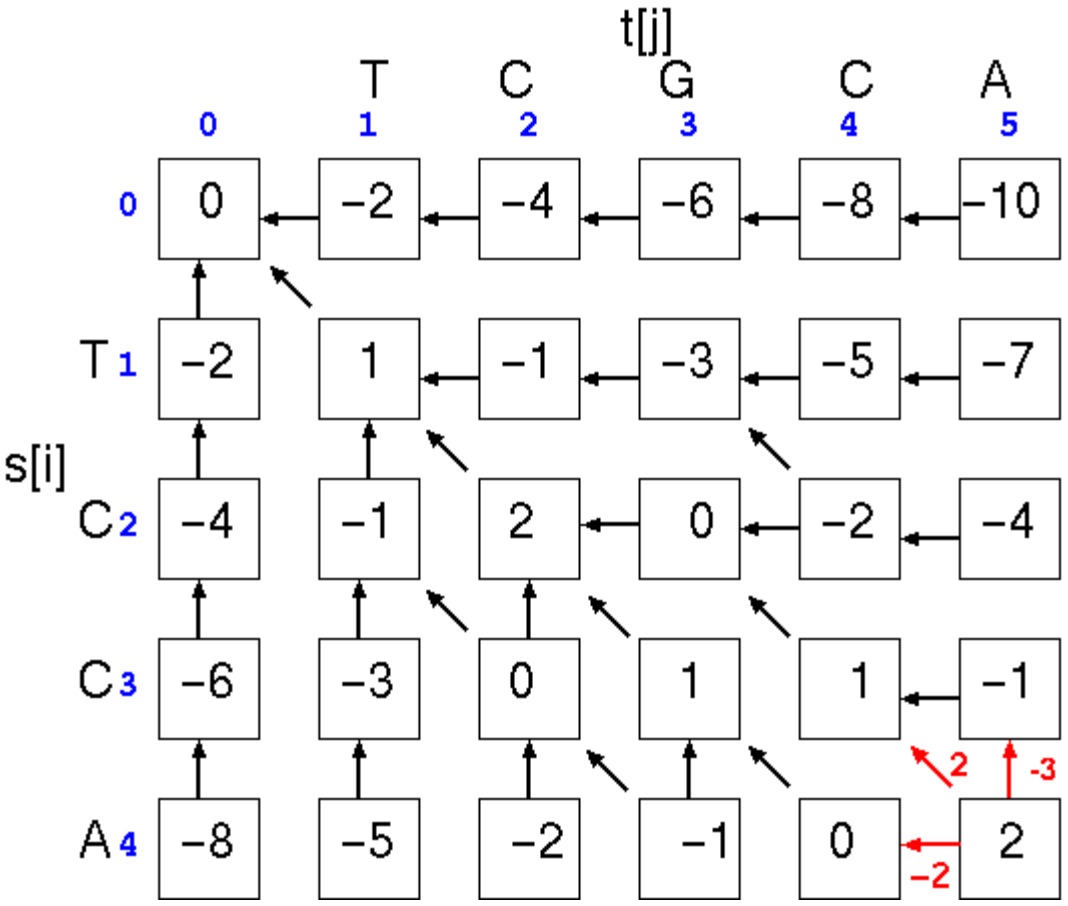
Dynamic programming: example



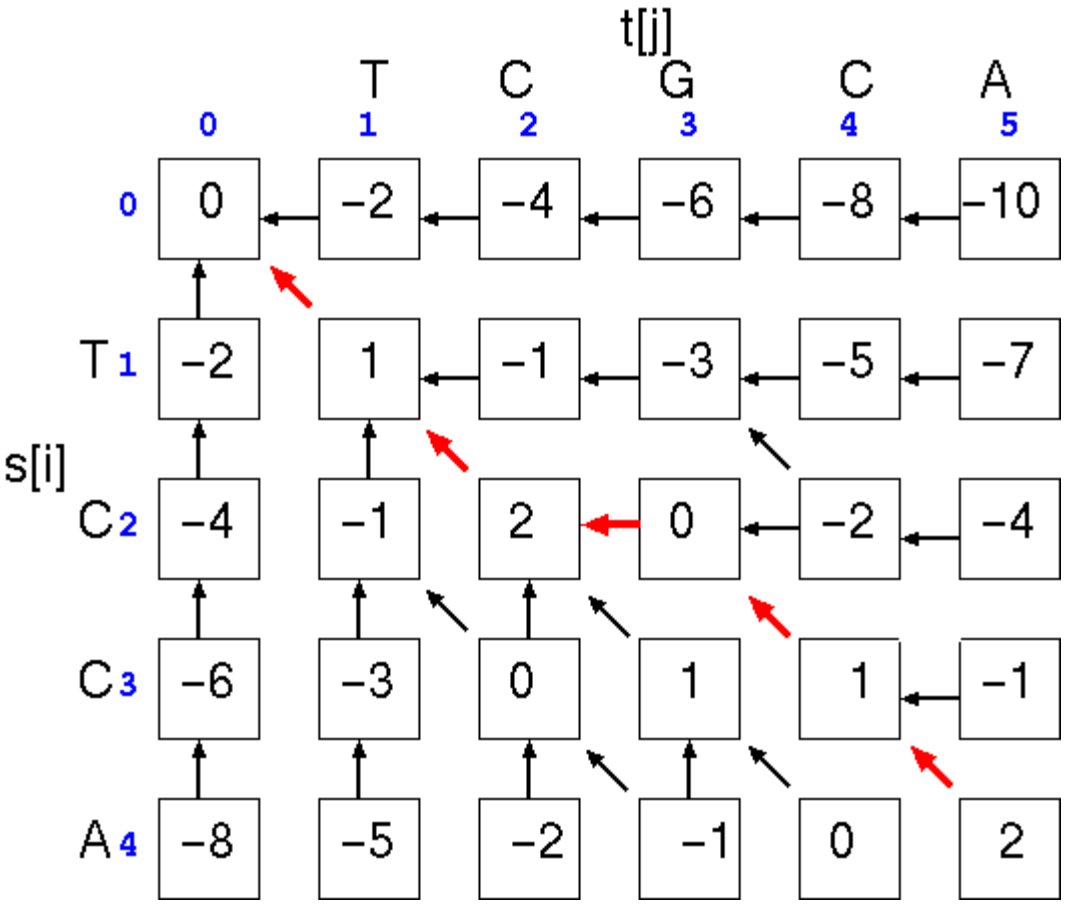
Dynamic programming: example



Dynamic programming: example



Dynamic programming: example

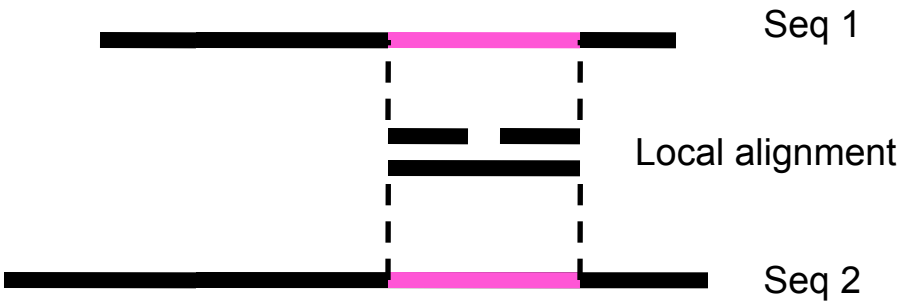



```
T C G C A
: :   : :
T C - C A
-------------
1+1-2+1+1 = 2
```

Global versus local alignments

Global alignment: align full length of both sequences.
(The “Needleman-Wunsch” algorithm).

Local alignment: find best partial alignment of two sequences
(the “Smith-Waterman” algorithm).



Local alignment overview

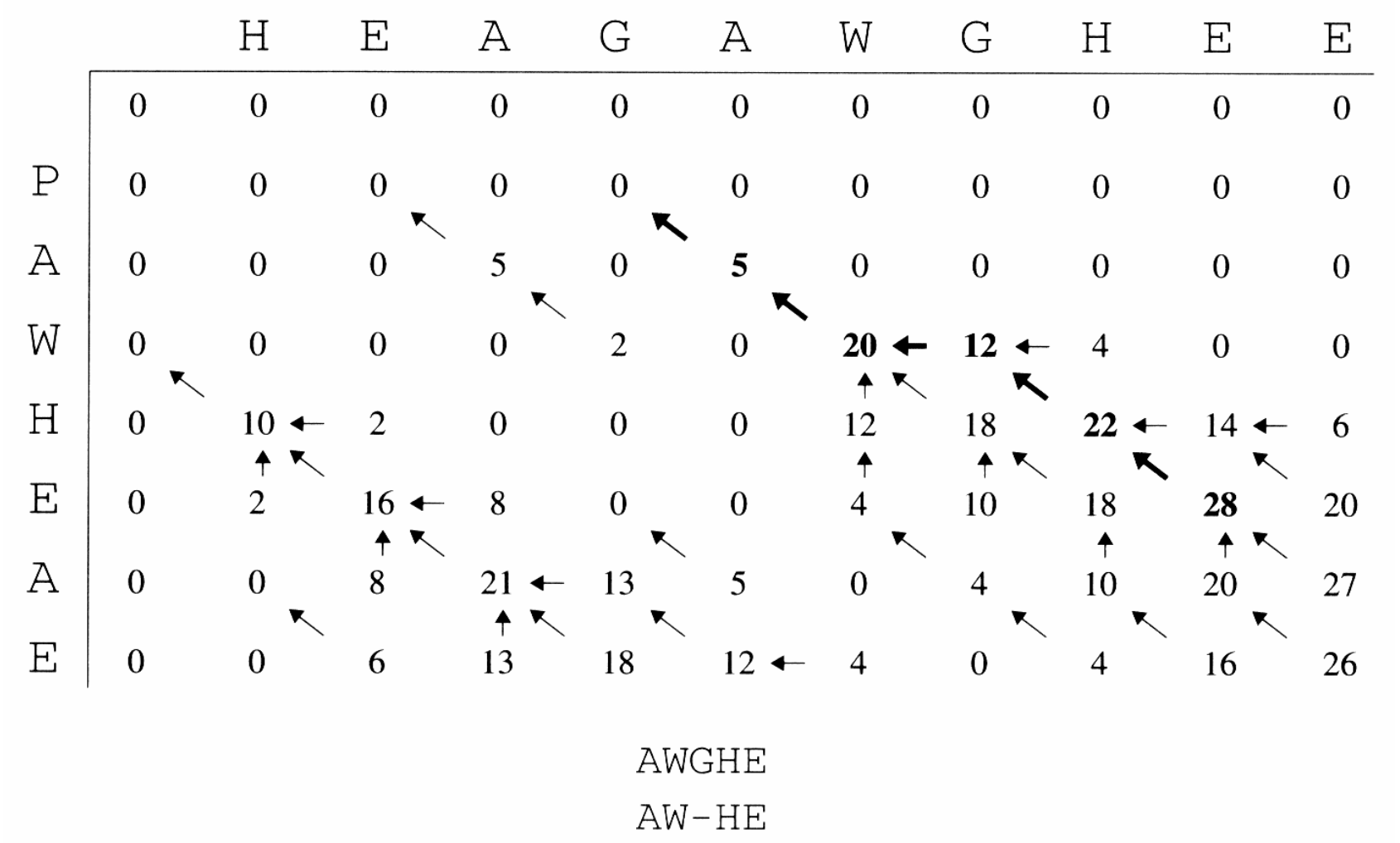
• The recursive formula is changed by adding a fourth possibility: zero. This means local alignment scores are never negative.

$$\text{score}(x,y) = \max \begin{cases} \text{score}(x,y-1) - \text{gap-penalty} \\ \text{score}(x-1,y-1) + \text{substitution-score}(x,y) \\ \text{score}(x-1,y) - \text{gap-penalty} \\ 0 \end{cases}$$

• Trace-back is started at the highest value rather than in lower right corner

• Trace-back is stopped as soon as a zero is encountered

Local alignment: example



Substitution matrices and sequence similarity

- Substitution matrices come as series of matrices calculated for different degrees of sequence similarity (different evolutionary distances).

- ”Hard” matrices are designed for similar sequences
  - Hard matrices a designated by high numbers in the BLOSUM series (e.g., BLOSUM80)
  - Hard matrices yield short, highly conserved alignments

- ”Soft” matrices are designed for less similar sequences
  - Soft matrices have low BLOSUM values (45)
  - Soft matrices yield longer, less well conserved alignments

Alignments: things to keep in mind

"Optimal alignment" means "having the highest possible score, given substitution matrix and set of gap penalties".

This is NOT necessarily the biologically most meaningful alignment.

Specifically, the underlying assumptions are often wrong: substitutions are not equally frequent at all positions, affine gap penalties do not model insertion/deletion well, etc.

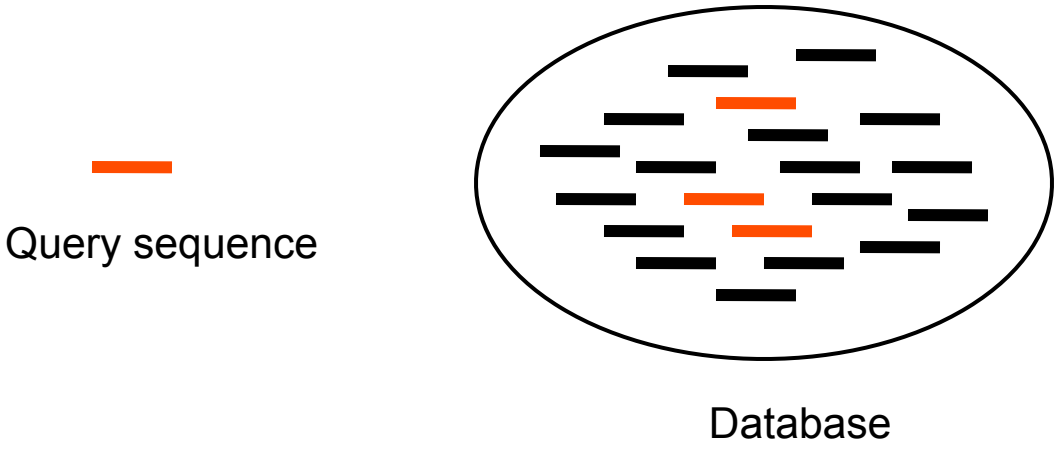
Pairwise alignment programs always produce an alignment - even when it does not make sense to align sequences.

BLAST

Anders Gorm Pedersen
&
Rasmus Wernersson

Database searching

Using pairwise alignments to search databases for similar sequences



Database searching

Most common use of pairwise sequence alignments is to search databases for related sequences. For instance: find probable function of newly isolated protein by identifying similar proteins with known function.

Most often, ***local*** alignment (“Smith-Waterman”) is used for database searching: you are interested in finding out if ANY domain in your protein looks like something that is known.

Often, full Smith-Waterman is too time-consuming for searching large databases, so heuristic methods are used (fasta, BLAST).

Database searching: heuristic search algorithms

FASTA (Pearson 1995)

Uses heuristics to avoid calculating the full dynamic programming matrix

Speed up searches by an order of magnitude compared to full Smith-Waterman

The statistical side of FASTA is still stronger than BLAST

BLAST (Altschul 1990, 1997)

Uses rapid word lookup methods to completely skip most of the database entries

Extremely fast

- One order of magnitude faster than FASTA
- Two orders of magnitude faster than Smith-Waterman

Almost as sensitive as FASTA

BLAST flavors

BLASTN
- Nucleotide query sequence
- Nucleotide database

BLASTP
- Protein query sequence
- Protein database

BLASTX
- Nucleotide query sequence
- Protein database
- Compares all six reading frames with the database

TBLASTN
- Protein query sequence
- Nucleotide database
- ”On the fly” six frame translation of database

TBLASTX
- Nucleotide query sequence
- Nucleotide database
- Compares all reading frames of query with all reading frames of the database

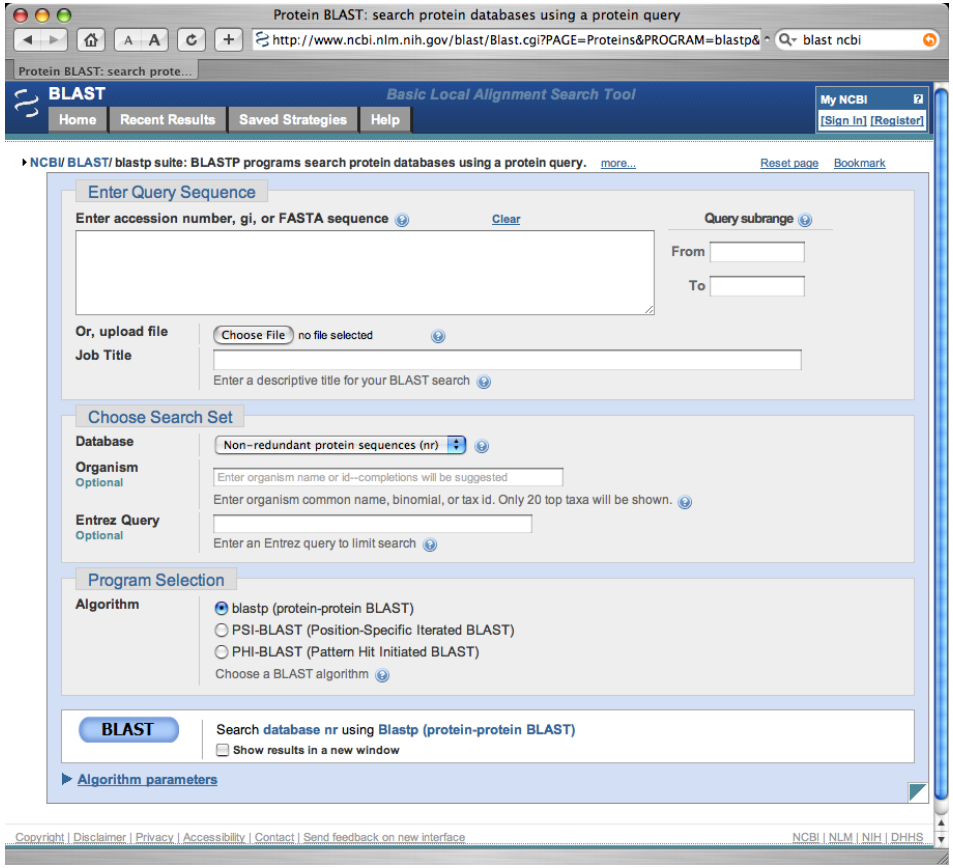
Searching on the web: BLAST at NCBI

Very fast computer dedicated to running BLAST searches

Many databases that are always up to date (e.g. NR and Human Genome

Nice simple web interface

But you still need knowledge about BLAST to use it properly



When is a database hit significant?

- **Problem**:
  - Even unrelated sequences can be aligned (yielding a low score)
  - How do we know if a database hit is meaningful?
  - When is an alignment score sufficiently high?

- **Solution**:
  - Determine the range of alignment scores you would expect to get for random reasons (i.e., when aligning unrelated sequences).
  - Compare actual scores to the distribution of random scores.
  - Is the real score much higher than you'd expect by chance?

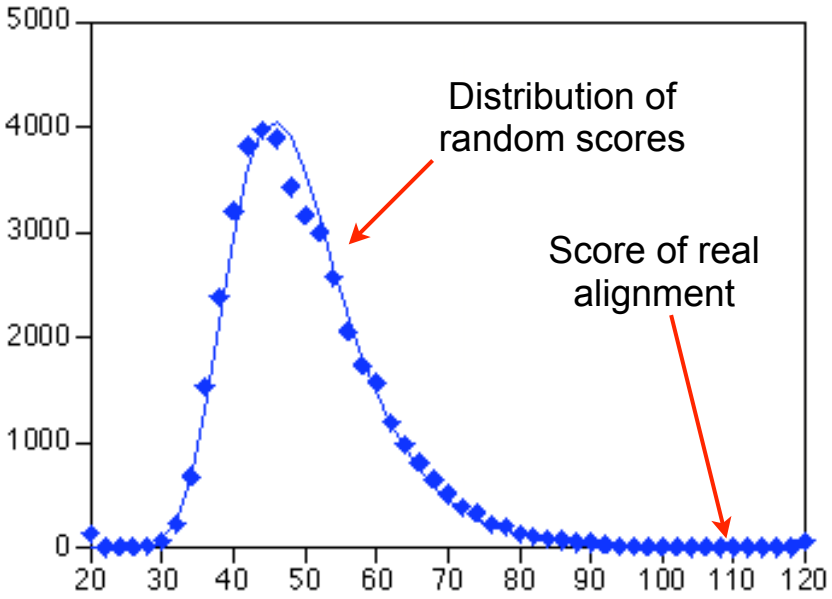
Distribution of random alignment scores

- **Software simulation**

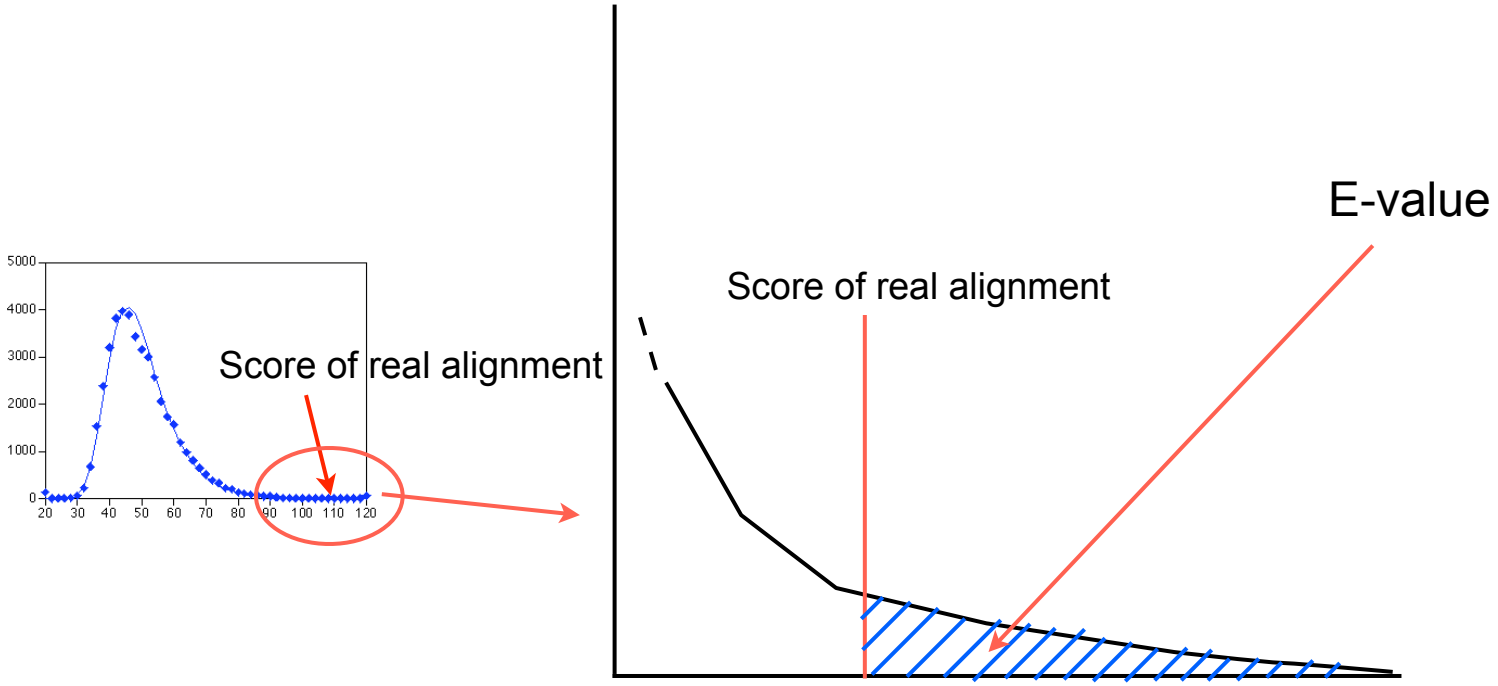
Significance of alignment score expressed as E-value

Searching a database of **unrelated** sequences results in scores following an extreme value distribution

The exact shape and location of the distribution depends on the exact nature of the database and the query sequence



Significance of alignment score expressed as E-value



E-value: the number of random hits with score >= real score

Want E-values below 1 (the lower the better)

BLAST heuristics

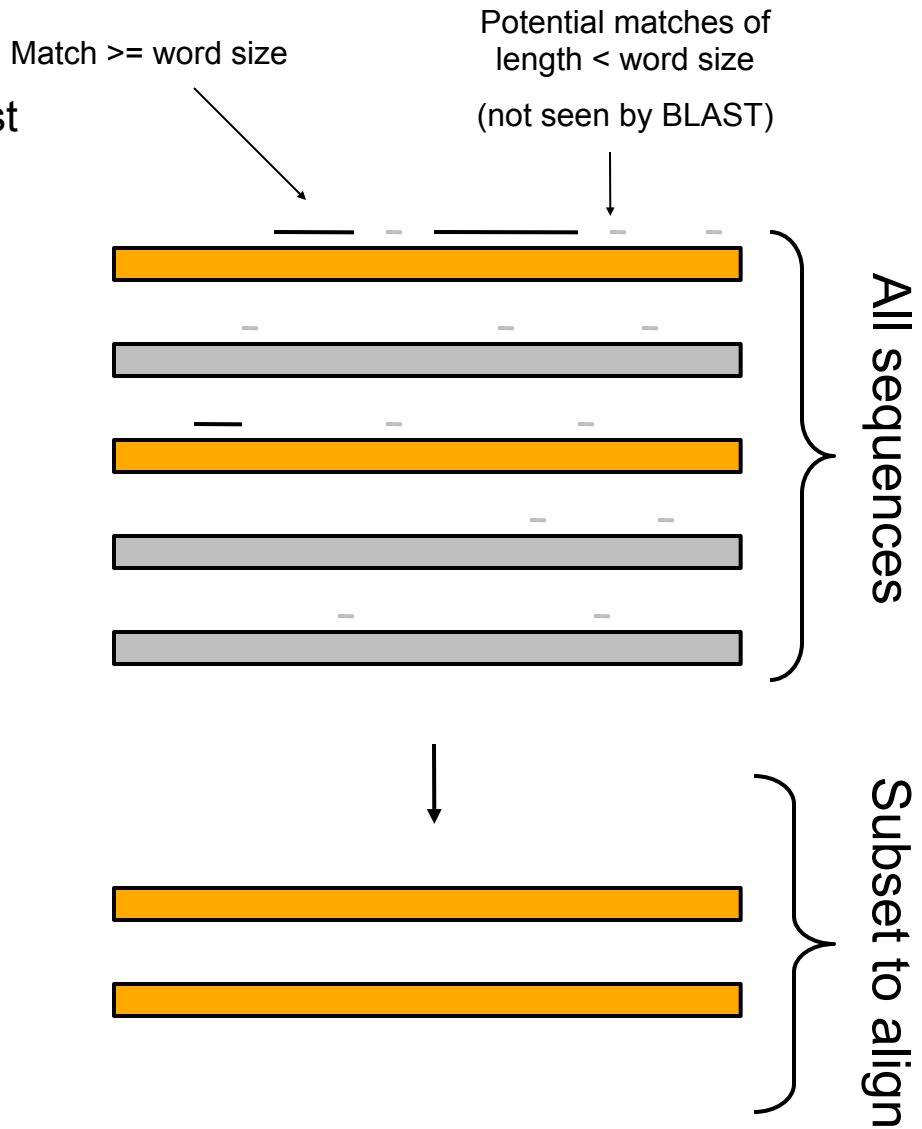
- BLAST speeds up the search >100x by pre-screening the database sequences and only performing the full Dynamic Programming on "*promising*" sequences.

- Promising sequences: database sequences that have sub-strings ("words") which also occur in the query sequence (found rapidly using a so-called "**suffix-tree"**)

- **BLASTN** and **BLASTP** use different criteria for overlap required for a sequence to be deemed promising

BLASTN

- Heuristics:
  - Perfect match “word” of at least size: 7, 11 (default) or 15.
- Alignment matrix:
  - Match: **1**
  - Mismatch: **-3**
- Notice: All mismatches are equally penalized:
  - E.g. A:G == A:C == A:T
  - More advanced models for DNA evolution do exist.



BLASTP

- Heuristics:
  - 2 x “Near match” within a window.
  - Default word length: 3 aa
  - Default window length: 40 aa
- Alignment matrix:
  - PAM and BLOSUM-series (default: BLOSUM 62)
- Notice: These alignment matrices incorporate knowledge about protein evolution.